

# Tarjousjärjestelmän määrittely

## - Case KK-trade

Järvinen, Markku; Tilander, Jesse

2014 Leppävaara

Laurea-ammattikorkeakoulu  
Laurea Leppävaara

## Tarjousjärjestelmän määrittely

- Case KK-trade

Järvinen, Markku; Tilander, Jesse  
Tietojenkäsittely  
Opinnäytetyö  
Kesäkuu, 2014

Järvinen, Markku; Tilander, Jesse

### Tarjousjärjestelmän määrittely - Case KK-trade

Vuosi 2014

Sivumäärä 26

Toiminnallisen opinnäytetyön tarkoitus oli määrittää asiakkaalle tarjousjärjestelmä käyttäen eri työkaluja. Yritys toivoi saavansa toimivan ja helppokäyttöisen tarjousjärjestelmän, joka mahdollistaisi asiakkaiden tarjouspyyntöjen suoran toimituksen yritykselle. Asiakasyritys toivoi CMS (engl. CMS, Content Management System) sisällönhallintajärjestelmän implementointia sivulle sen helpon käytettävyyden takia. Sisällönhallintajärjestelmä muodostaa internet-sivun teknisen puolen, jolla ohjataan pääsivun toimintaa. Oli tärkeää saada määritettyä järjestelmä, jota yritys pystyisi hyödyntämään pitkälle tulevaisuuteen.

Opinnäytetyön tärkein tavoite oli, miten tehdään tietokantajärjestelmä pienine lisukkeineen hyvin ja oikeaoppisesti. Asiakas halusi tietokannan, josta näkee heidän maahantuontituotteet, josta saadaan valittua tuotteet tarjouspyyntöpohjalle ja mahdollisesti laskettua katteet tuotteista valmiiksi. Tästä ne menisivät suoraan myyjälle, joka pystyisi hyväksymään tarjouspyynnön ja lähettämään tarjouksen suoraan tehtaalle englanniksi käännettynä.

Tietoperustaan valittiin ohjelmat sekä käsitteet, joita käytimme demon luontiin ja CMS Made Simple ohjelma asiakkaan toiveesta. Näillä oli tarkoitus hankkia tietoa sekä kokemusta demon toteutukseen ja tietämys jatkokehitykseen. Tietoperustassa otettiin huomioon asiakkaan toimeksiannossa määritellyt tarpeet ja haluttu lopputulos. Lähteet valittiin sen perusteella, mitkä antoivat paremman ymmärryksen ohjelmista sekä eduista niiden käytössä.

Suunnitteluvaiheessa käytiin läpi asiakkaan toiveet ja miten edetä, kun olimme saaneet tarvittavan tietoperustan hankittua. Suunniteltiin tietokanta käyttäen UML-kaaviota. Lisäksi toteutettiin demo, jonka tarkoitus oli antaa lisänäkemyksiä, miten kokonaisuuden tulisi toimia. Demossa näytetään eri vaiheet: sisäänkirjautuminen, tuotteitten valinta tilauslomakkeesta, katteen laskeminen katelaskurilla ja tietokanta.

Opinnäytetyössä opittiin ajankäytön hallintaa, asiakkaan kanssa kommunikointia sekä toteutuksessa käytetyn tietoperustan hyödyntämistä että ohjelmien etujen hyödyntämistä. Työssä opittiin tarjousjärjestelmän toteuttaminen suunnitteluvaiheesta määrittelyyn sekä demossa käytännön tasolla, miten tulisi toteuttaa asiakkaan tarjousjärjestelmä.

Järvinen Markku; Tilander Jesse

### Defining a Quoting System - A Case Study KK-Trade

Year	2014	Pages	26
------	------	-------	----

The purpose of this thesis was to define premises for a working bargaining system to facilitate the customer's import company's work load and to make business more fluent. The company that assigned this project had some requests about the system. One of the requests was about the usage of a management tool called CMS Made Simple, which they were familiar with. CMS is short for Content Management System, which is a web based management tool for managing content inside the webpage.

The main objective of this thesis was to acquire knowledge about creating a system which works and satisfies the customer needs and purposes. The customer wanted a database showing the products that can be selected from the page and the products would be added to the bargaining system layout page. The profit margin could be counted on the page. From this page you could send the offer straight to the import manager, who would accept, deny or modify the offered deal. After this the offer could be translated into English and sent straight to the factory for processing.

The knowledge base for the thesis is based on programs and program languages which were used in the creation of a demo for this project. At the customer's request a decision was made to include the CMS Made Simple program in the knowledge base for future development. Before acquiring the knowledge base for this thesis the customer's needs and purposes had to be taken into account. References for this were selected based on this and what programs were chosen to aid in the creation of the demo.

The planning phase began after obtaining the right knowledge base. For the database planning a UML diagram was created in order to aid to gain a better picture of the database functionality and a demo was created to demonstrate a possible way to create the database. The demo presents the logging in, selection of the products on the order form, counting the profit margin and the database.

The thesis taught how to better manage time resources, communication skills, and how to use the knowledge base acquired when implementing it. The project taught how to create a bargaining system from planning to defining and from the demo to the practical side of implementing and creating the database itself.

Keywords      Quoting system, Database, PHP, MySQL, Webpage, Developing, Functionality

## Sisällys

1	Johdanto .....	6
1.1	Toimeksiantaja .....	6
1.2	Tavoitteet .....	6
1.3	Menetelmien käyttö .....	7
1.4	Edellytykset onnistumiseen .....	7
2	Toteutus .....	8
2.1	PHP .....	8
2.2	SQL .....	8
2.3	CMS .....	9
2.4	JavaScript .....	10
3	Ohjelma .....	10
3.1	CMS Made Simple .....	10
3.2	XAMPP .....	10
3.3	MySQL .....	11
3.4	NetBeans IDE .....	11
4	Suunnittelu .....	11
5	Tietokantasuunnittelu ja määrittely .....	12
5.1	Käyttäjäryhmät .....	12
5.2	Tietokannansisältö .....	12
5.3	UML-kaavio .....	13
6	Mahdollinen toteutussuunnitelma .....	14
6.1	PHP ja MySQL .....	14
6.2	Demon toteutus .....	15
7	Jatkokehitysmahdollisuudet .....	17
7.1	Sivuston kehitys .....	17
7.2	Mobiiliratkaisut .....	17
8	Yhteenveto .....	19
	Lähteet .....	21
	Kuvat 22	
	Liitteet .....	23

## 1 Johdanto

Opinnäytetyön tavoitteena on määritellä ja suunnitella ferrariagri.fi-verkkosivuille nykypäiväinen tarjousjärjestelmä, joka vastaa asiakkaan KK-Trade määrittelyjä ja tarpeita. Asiakas tarvitsee kyseisen järjestelmän kasvattaakseen, parantaakseen ja helpottaakseen yritystoimintaa.

Tarjousjärjestelmällä tässä työssä tarkoitetaan tietokantaa, jossa on tuotteet listattuna ja hinnoiteltuna. Tietokannasta saadaan tuotteet ja niiden hinnat sekä katteet laskettua ja tulostettua tarjoukseen.

Työhön tehdään ”demo” eli demonstraatio. Demossa esitellään, kuinka ohjelman voisi toteuttaa. Demo esittää yhden mahdollisuuden ohjelman toteutuksesta helposti ja pragmaattisesti. Demo sisällytetään työhön kuvankaappauksin. Kuvankaappauksissa havainnollistetaan ohjelman perustoimintoja sekä rakennetta.

Työssä käytämme koulutuksessa opittua tietotaitoa ja työelämässä saavutettua vankkaa kokemusta aiheeseen liittyen. Työssä haasteellisimmaksi näemme työn toteutuksena syntyvän sovelluksen sopivuuden asiakkaan tarpeisiin sekä yhteistyön muodostamisen asiakkaan ja asiakkaan sidosryhmien välille.

### 1.1 Toimeksiantaja

Projektinvetäjä on digitaali.fi, joka suunnittelee ja toteuttaa laadukkaita web- ja mainospalveluita ja toimeksiantajana toimii KK-trade Oy, jonka toimintaan sisältyy suuren koneiden ja niiden varaosien maahantuontia sekä jälleenmyyntityö. KK-trade toimii ferrariagri:n maahantuojana Suomessa. KK-trade:n toimipiste löytyy Savonlinnasta.

### 1.2 Tavoitteet

Tavoitteet ovat määritellä tarjouksiin perustuva tietokantajärjestelmä yrityksen käyttöön. Jonka pohjalta asiakas pystyy arvioimaan tarjouksen laadun, kulut sekä tuottavuuden. Työssä on tavoitteena luoda määrittely tarjousjärjestelmälle. Määrittely tukee ohjelman luontivaihetta. Tarjousjärjestelmän tarkoituksena on helpottaa asiakas yritysten tässä tapauksessa jälleenmyyjä yritysten tilausten luontia, sekä lähetystä maahantuojalle. Maahantuoja on yritys, joka on työn alkuperäinen toimeksiantaja (KK-Trade oy). Lisäksi tavoitteena on pohtia mahdollisia tulevaisuuden kehitys sekä parannus ratkaisuja.

Tarkoituksena on parantaa ongelmanratkaisutaitoja sekä oppia entistä enemmän, kuinka toimiva SQL-pohjainen PHP-sivusto rakennetaan oikeaoppisesti alusta alkaen itse, ilman turvautumista valmiiksi olemassa oleviin tuotteisiin, joita on tarjolla lukuisia. Työssä käytetään niin Laurea-ammattikorkeakoulussa opittuja asioita kuin omaa valmista osaamista. Lisäksi työn eri vaiheissa halutaan paikantaa mahdollisia ongelmatilanteita ja ennaltaehkäistä niiden syntymistä myöhemmässä vaiheessa.

### 1.3 Menetelmien käyttö

Työssä käytetään "Top-down"-menetelmää. Menetelmä sisältää neljä vaihetta työn suunnittelu, tehdyn suunnitelman analysointi, määrittely analysoidusta suunnitelmasta ja lopullinen toteutus testauksineen. Koska työ sisältää vain suunnittelun ja määrittelyn käytetään vain kahta ensimmäistä vaihetta.

Toisena tutkimusmenetelmänä käytetään demoa, jolla demonstroidaan mahdollinen toteutus, toimivuus ja mahdollinen lopputulos käytännön esimerkin avulla. Sillä pyritään avaamaan ohjelman mahdollisuuksia kehitykseen ja tulevaan käyttöön.

### 1.4 Edellytykset onnistumiseen

Opinnäytetyön onnistumiseen vaaditaan tarkkaa aikataulutusta sekä jatkuvaa yhteistyötä KK-trade Oy:n ja Digitaali.fi:n kanssa. Yhteistyö asiakkaiden kanssa tapahtuu sähköpostin, virtuaalipalavereiden ja henkilökohtaisten tapaamisien merkeissä.

## 2 Toteutus

Työn toteutus rakentuu vahvasti ohjelmointi kieliin. Toteutuksen kannalta tärkeimmät komponentit sivuston dynaamisuuteen ovat PHP ja MySQL-relaatiotietokannat. Apuna työssä toimi NetBeans IDE, joka teki koodin luonnista ja muokkaamisesta helpompaa. Demon toteutuksessa pienessä osassa oli myös JavaScript, millä luotiin "log-in" pop up-tietovarmennus.

### 2.1 PHP

Aluksi PHP koostui joukoista makroja, jotka autoivat koodaajaa ylläpitämään henkilökohtaisia kotisivuja, tästä nimi PHP (Personal Homepages). Sen jälkeen PHP:n ominaisuudet ovat lisääntyneet ja PHP:stä tuli täysin oma ohjelmointikieli, jolla voidaan hallita jopa laajoja tietokantapohjaisia online-ympäristöjä. (Zandstra 2001, 20)

PHP on hyvä valinta sen laajuuden ja tuen takia. PHP on open source-tuote, joka takaa sitoutuneen käyttäjäyhteisön. PHP-koodia voidaan suorittaa useimmissa käyttöjärjestelmissä ja palvelimissa. Useiden projektien tuotantoaika lyhenee verrattuna muihin skriptikieliin juuri tämän ansiosta. (Zandstra 2001, 21)

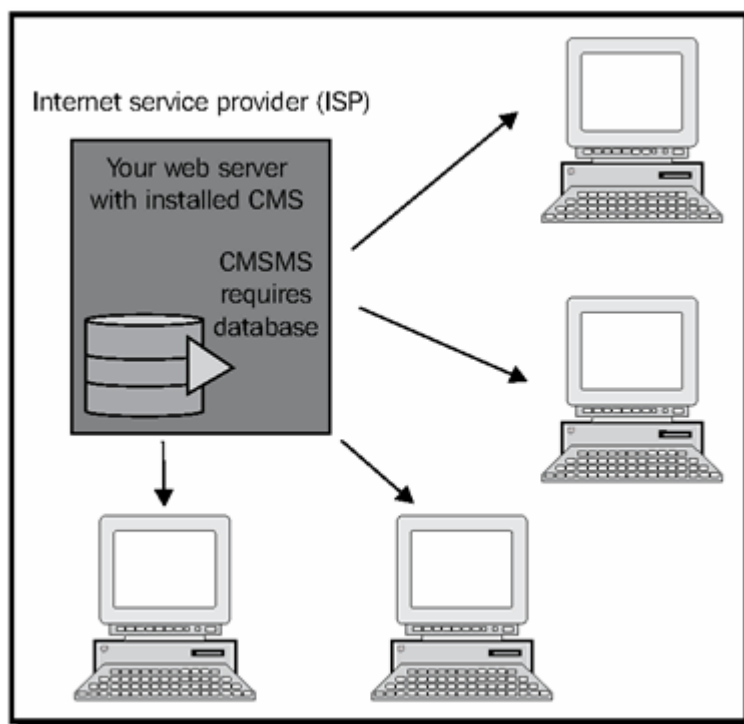
### 2.2 SQL

SQL (Structured Query Language) on ohjelmointikieli, joka on luotu relaatiotietokantojen luomiseen. SQL-ohjelmointikielellä pystyy hallitsemaan ja luomaan tietokantoja erilaisten ohjelmien kautta, kuten MySQL, MariaDB yms. (SQL in a Nutshell Kline, Kline, Hunt. (s.1)) SQL:llä voidaan määritellä tietokannan käyttäjät ja heidän käyttöoikeutensa, tietokannan tietosisältö ja hakea tietoa tietokannasta. Tiedon hakeminen tietokannasta pystytään näyttämään näytöllä, tiedostona sekä sovellusohjelmalla. Tietokantaa voidaan päivittää vuorovai-  
kutteisesti sekä sovellusohjelman kautta.



## 2.3 CMS

CMS (Content Management System) tarkoittaa sisällönhallintajärjestelmää. Yleisesti sisällönhallintajärjestelmää käytetään sivuston rakenteen luontiin ja ylläpitoon. Sisällönhallintajärjestelmän tärkeimpiä tehtäviä on rakenteen, ulkonäön ja ohjelmoinnin jaottelu. (Kuva 1: CMS Toimintakaavio)



Kuva 1: CMS Toimintakaavio

Sisällönhallintajärjestelmä on luotu käyttäjälle, joka ei erikseen tarvitse ohjelmointitaitoa. Käyttäjä käyttää graafista käyttöliittymää luodessaan haluamansa materiaalin. (CMS Made Simple 1.6 Beginner's Guide, Hauschildt. (s.8))

Sisällönhallintajärjestelmää (CMS) ei kannata sekoittaa verkkosivujen luontiohjelmiin, koska sisällönhallintajärjestelmä ei tätä ole. Verkkosivujen luontiohjelmat ovat niitä varten, jotka haluavat luoda verkkosivut tietämättä teknisestä osa-alueesta. Tämä tekee verkkosivut rajallisiksi ja pienille käyttäjämäärille sopiviksi. Useimmiten verkkosivujen luontiohjelmissa tehdyt sivut on tarkoitettu yksityishenkilöiden kotisivuiksi. Sisällönhallintajärjestelmä antaa mahdollisuuden luoda täysin muokattavissa olevan verkkosivukokonaisuuden ja ympäristön. (CMS Made Simple 1.6 Beginner's Guide, Hauschildt. (s.9))

## 2.4 JavaScript

JavaScript on Netscapen luoma web-ympäristöissä yleinen komentosarjakieli. JavaScript mahdollistaa dynaamisten ominaisuuksien luonnin web-sivustolle. Nykypäivänä kieli on yksi käytetyimmistä web-kehityssaralla.

## 3 Ohjelma

Seuraavilla ohjelmilla toteutettiin kyseisen työn demo-osuus. Kyseiset työkalut sopivat myös demon rakentamisen jatkamiseen aina käyttöönottoon asti. Kappaleessa esitellään myös ohjelma, mitä ei demossa käytetty CMS Made Simple, joka valittiin huomioon ottaen kyseisen ohjelman hinta sekä asiakkaan aikaisempi kokemus ohjelman käytöstä.

### 3.1 CMS Made Simple

CMS Made Simple on ilmainen open source-tuote, joka on kehitetty CMS:stä ja luotu helpoksi käyttää. CMS:ssä on mukana valmiina rakennettuja ominaisuuksia. CMS Made Simple:n suurimpia hyötyjä ovat joustavuus sekä hyvin luotu tuki HTML- ja CSS-koodikielille. CMS Made Simple:en pystytään lisäämään satoja moduuleita, jotka mahdollistavat persoonallisen käyttömukavuuden ja käytettävyyden.

CMS Made Simple on ilmaisen lähdekoodin ohjelma, jonka takia sen käyttö on ilmainen yksityiseen sekä kaupalliseen käyttöön. Sen takia ohjelmalla on laaja yhteisötuki. Laajan yhteisötuen ansiosta ohjelmalla on kehitetty useita laajennuksia, korjauksia sekä moduuleita. (CMS Made Simple 1.6 Beginner's Guide, Hauschildt. (s.9))

### 3.2 XAMPP

XAMPP on ilmainen open source-alustan palvelintyökalu. Josta löytyy valmiiksi paketoituna MySQL, PHP, Perl ja Apache Web Server. XAMPP sisältää myös erilliset SMTP- ja FTP-palvelimet.

XAMPP on helppo valinta monelle yritykselle. Järjestelmän suurimmat hyödyt ovat helppokäyttöisyys, yhteensopivuus lähes kaikkien palvelinympäristöjen kanssa, sekä järjestelmän avoimen lähdekoodin mahdollistamat alhaiset hankinta ja ylläpitokustannukset.

### 3.3 MySQL

MySQL on maailman tunnetuin tietokanta open source-tuote, joka on yleisesti käytössä PHP:n kanssa. MySQL tunnetuimmat käyttöympäristöt ovat Oracle- sekä MS SQL-palvelimet. MySQL:n yleisen sekä vapaanlähdekoodin ansioista se sisältää laajasti apuvälineitä, jotka mahdollistavat tietokantojen ylläpidon sekä muokkaamisen todella helposti ja vaivattomasti.

### 3.4 NetBeans IDE

NetBeans IDE (integrated development environment) on avoimenlähdekoodin ilmaisohjelma mikä on luotu helpottamaan ohjelmointia. Ohjelma toimii koodieditorina, mikä helpottaa virheiden tai ongelmatilanteiden havainnointia reaaliajassa. NetBeans on luotu pääsääntöisesti Javan ohjelmointiin mutta se tukee myös PHP, C/C++ ja HTML5. Työssä käytössä olivat pääosin PHP sekä Java.

## 4 Suunnittelu

Suunnitteluvaiheessa määritettiin tärkeimmät eli asiakkaan asettamat toiveet ja pyynnöt sekä tavat, miten toteuttaa niitä. Seuraavaksi siirryttiin tietokantasuunnitteluun, jolloin määriteltiin ohjelma, kieli, käytötavat ja säännöt.

Työ toteutettiin asiakkaan toiveiden mukaisesti PHP/MySQL:llä. Asiakkaan sisällönhallinnosta vastaa Digitaali.fi valitsi CMS Made Simple hallinnointiohjelman, mikä helpottaa sivuston sisällönhallintaa.

Asiakkaalla oli seuraavia toiveita. Asiakas halusi hinnaston, joka toimii kauppiaalla tarjouspohjana, joka on apuna tilauksia tehtäessä. Ohjelman tulisi myös estää mahdolliset ongelmat sekä ristiriitatilaukset. Asiakas piti myös tärkeänä, että tilauskaavake on mahdollista kääntää suomesta englanniksi. Toimeksiantajan toive oli myös saada mahdollisuus muuttaa jo täytettyä tilausta. Erikoisominaisuutena tilauskaavakkeen tilaajilla olisi mahdollisuus käyttää kate-laskuria, joka toimisi tilauskaavakkeen ulkopuolella. Tilaaja pystyisi laskemaan nopeasti tilauksensa katteen jo ennen tilausta.

## 5 Tietokantasuunnittelu ja määrittely

Tietokanta toteutetaan XAMPP:in kautta MySQL:llä, joka mahdollistaa helpon ja kätevän hallinnoinnin sekä muokkauksen. MySQL:n kautta luodaan käyttäjärhytät ja tietokannat. Asiakkaan kannalta tämä on hyvä, koska se mahdollistaa mutkattoman ja vaivattoman tavan hallita sekä ylläpitää sisällönhallintajärjestelmää.

Tietokanta on suljettu eli käyttäjältä vaaditaan käyttäjätunnus, jonka luo ylläpito. Tietokantaan tulee asiakkaan tuotevalikoima, joka jaetaan noin kuuteen eri ryhmään. Ryhmien erittely tapahtuu niiden omilla ID-tunnuksilla. Joiden avulla karsitaan pois mahdollisia virhevalintoja ja tämä helpottaa ylläpitäjien työtaakkaa.

### 5.1 Käyttäjärhytät

Käyttäjärhytät toteutetaan niin, että ylläpito luo käyttäjälle käyttäjätunnuksen, jonka jälkeen käyttäjä saa määrittellä itse tilinsä salasanan. Ylläpito määrittelee halutessaan käyttäjille lisäoikeuksia. Käyttäjärhytät määrittävät käyttäjille annettavat oikeudet. Luodulle käyttäjälle käyttäjärhytässä tulee sellaiset käyttöoikeudet, jotka mahdollistavat tietokannan tarkastelun ja tiedonhakemisen.

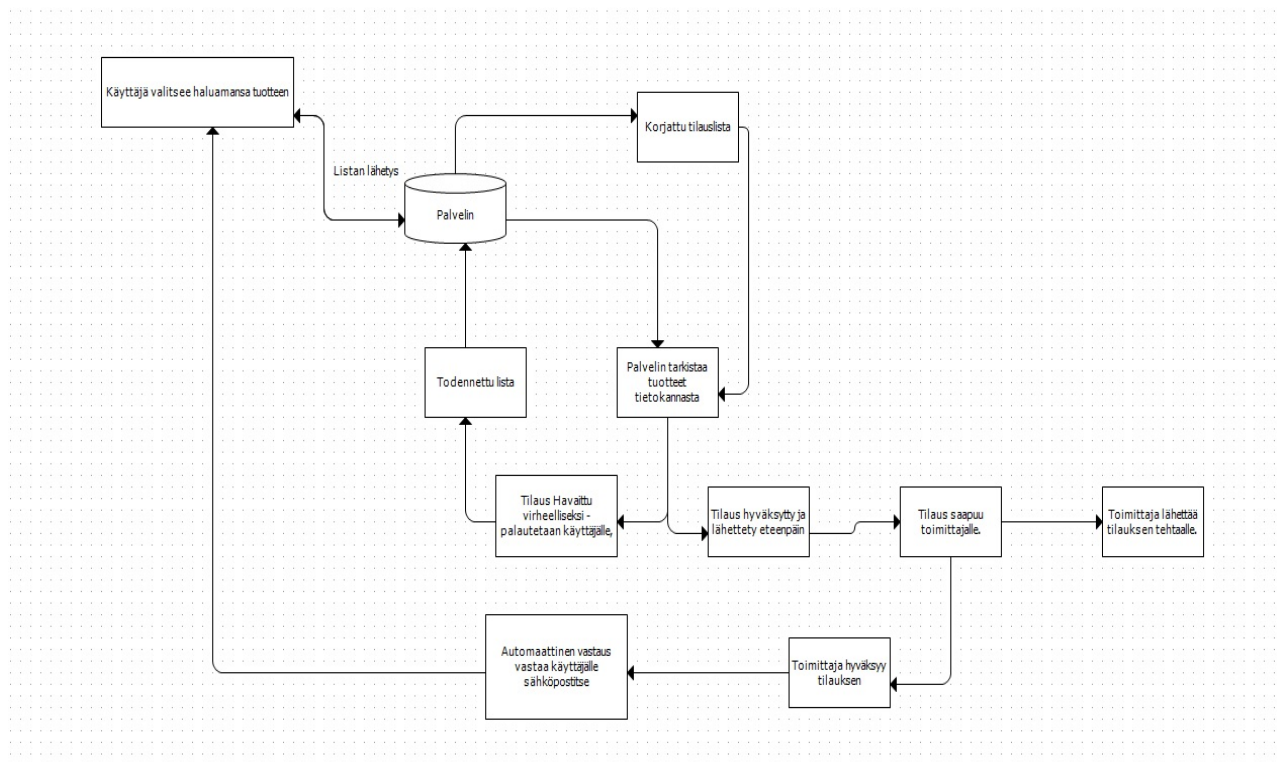
### 5.2 Tietokannansisältö

Työn tietokanta tulee sisältämään sääntöjä sekä useita taulukoita, jotka sisältävät yrityksen tuotetietoja. Näiden tuotetietojen perusteella tarjouksentekijä voi valita haluamansa tuotteet. Tietokannan puolella mahdolliset yhteensopivuusongelmat tuotteiden ja osien kanssa on estetty niin, että jokaiselle laitteelle on luotu oma yhteensopivuustaulukko, mikä sisältää tuotteen ja vain kyseiseen tuotteeseen yhteensopivat osat. Tämän toteutuksen ansiosta välttäneen "vikavalinnoilta".

Tietokanta sisältää myös taulukot käyttäjätunnuksista ja niiden salasanoista. Tämä toimii ensisijaisena tietoturvana tarjouspyyntöjärjestelmässä. Tällä pois suljetaan tai suljetaan pois mahdolliset asiattomat tarjouspyynnöt.

### 5.3 UML-kaavio

UML- kaaviolla kuvataan sivuston sekä tietokannan toimintaa hakua tehdessä. Kaaviolla kuvataan järjestelmän elinkaari tarjouksen luonnista lähetykseen.



Kuva 2: Tietokannan toimintakaavio

## 6 Mahdollinen toteutussuunnitelma

Tässä luvussa käsitellään suunnitteluvaiheen mahdollisuuksia sekä Demonstraation havainnoimia ratkaisuja. Demossa esitellään tehdyt ratkaisut ja näkemyksemme miten toteutus on mahdollista luoda. Luvussa avataan myös hieman mahdollisia tietoturvaongelmia ja miten niiltä on mahdollista välttyä.

### 6.1 PHP ja MySQL

PHP on oleellinen osa, jonka avulla sivut voidaan liittää tietokantaan. Liittämistä varten tehdään oma php-tiedosto. Tiedosto pitää sisällään MySQL\_connect-toiminnon, minkä tehtävänä on linkittää käyttäjä tietokantaan käyttäjätunnuksen ja salasanan avulla. Tiedosto pitää sisällään myös MySQL\_error-tarkistuksen mahdollisten virheiden välttämiseksi.

Ongelmana MySQL\_connect-toiminnossa on heikko tietoturva, joka mahdollistaa tiedonurkinan suojaamattomista paketeista. MySQL\_connect-toiminto siirtää kaiken datan suojaamattomassa tekstimuodossa, mikä tekee datan kaappauksesta sekä urkinnasta todella helppoa ja yksinkertaista. Mikäli tältä tietoturva uhalta halutaan välttyä, on syytä käyttää MySQL\_CLIENT\_SSL flag-toimintoa. Toiminto salaa datan salausalgoritmilla, joka tekee tiedonkaappauksesta sekä käytöstä huomattavasti haastavampaa. (Rook, 2010).

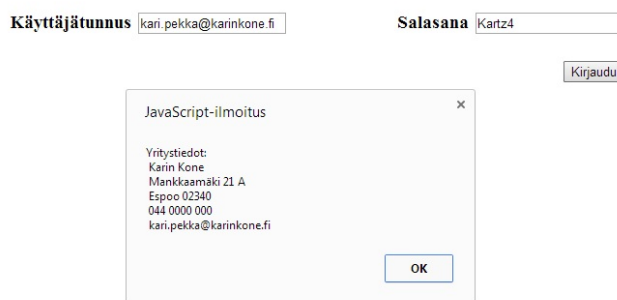
Kun tietokanta on tehty, aloitetaan erillisten php-tiedostojen luominen. PHP-tiedostojen tehtävänä on johtaa sivun dynaamista kulkua MySQL-tietokannassa. Toiminta perustuu PHP-lomakkeiden ja niihin syötettyihin tietoihin, jotka lähetetään tietokantaan. Prosessissa salana salattiin tietokantaa varten MD5-metodilla.

Toteutetussa demossa PHP toimii lomakkeiden luonnissa. Työn toteutusvaiheessa luodaan yksi lomake, mikä pitää sisällään asiakkaan täytettävän tilauslomakkeen ja "laskurin". Laskuri laskee katteen, joka toteutetaan yksinkertaisella laskentakaavalla. Laskurin tarkoitus on helpottaa asiakkaan ja toimittajan työmäärää; laskemalla katteen syötetyistä luvuista.

## 6.2 Demon toteutus

Demon luominen aloitettiin tekemällä XAMP:iin tietokannat tunnuksille sekä tuotteille. (Liite 4: Tietokanta) Tämän jälkeen aloitettiin itse sivujen luonti. Ensimmäiseksi luotiin sisäänkirjautumissivu, jonka tarkoituksena on tunnistaa, suojata ja todistaa käyttäjä. Sisäänkirjautumissivua luodessa käytettiin PHP:tä, CSS:ää, sekä Javascriptiä. Sisäänkirjautumissivulla käyttäjä laittaa tunnuksensa sekä salasansa avoimiin kenttiin. Sisäänkirjautuessa PHP lähettää todennuspyynnön tietokantaan, mikäli tietokanta vastaa ja myöntää luvan käyttäjälle edetä. (Liite 1: Sisäänkirjautuminen)

# SISÄÄNKIRJAUTUMINEN



The image shows a web login interface. At the top, there are two input fields: 'Käyttäjätunnus' (Username) containing 'kari.pekka@karinkone.fi' and 'Salasana' (Password) containing 'Kartz4'. To the right of the password field is a 'Kirjaudu' (Login) button. Below these fields is a JavaScript alert box titled 'JavaScript-ilmoitus' (JavaScript notification). The alert box contains the following text: 'Yritystiedot: Karin Kone, Mankkaamäki 21 A, Espoo 02340, 044 0000 000, kari.pekka@karinkone.fi'. There is an 'OK' button at the bottom right of the alert box.

Kuva 3: Sisäänkirjautuminen

Sisäänkirjautumisen jälkeen aukeaa tilauslomake. (Liite 2: Tilauslomake) Tilauslomakkeessa käyttäjä näkee ensimmäiseksi omat tilaustietonsa, jonka jälkeen käyttäjä pääsee tekemään tilaustansa ja valitsemaan haluamansa tuotteet dropdown-valikoista. Asiakkaan valittuaan haluamansa tuotteet voi asiakas lähettää tilauksen. Asiakas voi halutessaan lisätä tilaukseen lisätietoa. Sivulta löytyy myös katelaskuri, minkä saa avattua painamalla katelaskuri painiketta. (Liite 4: Katelaskuri) Asiakkaan painaessaan lähetä-painiketta tilaus lähtee sähköpostilla myyjälle.

**Tilauslomake**

Katelaskuri

**Asiakkaan tiedot:**

Karin Kone  
Mankkaamäki 21 A  
Espoo 02340  
044 0000 000  
kari.pekka@karinkone.fi

**Tilaus:**

Suurkone mallit	Osat
Raptor	Osa 4
Suurkone mallit	Osat
Cobram AR	Osa 5
Suurkone mallit	Osat
Cobram 60 RS	Osa 4

**Lisätiedot:**

Lähetä

Kuva 4: Tilauslomake



## 7 Jatkokehitysmahdollisuudet

Kyseisessä luvussa havainnoidaan jatkokehitysmahdollisuuksia sivuston kehityksen suhteen, sekä tulevaisuuden mahdollisuuksia mobiili-laitteille. Sivuston mahdollisia kehityssuuntia poh-tiessamme havainnoimme tulevaisuuden tietoturva-aasteet ja niiden tärkeyden. Toinen sivus-ton kehityksessä esille tullut asia on sivuston käyttäjäystävällisyyden parantaminen.

Jatkokehitysmahdollisuudet mobiili-laite saralla ovat hyvin laajat. Mobiili-alan "nuoruuden" takia alalla on mahdollisuuksia loputtomasti. Tulevaisuuden kehityksen suunta kertoo, mitä mahdollisuuksia kyseinen teknologia pitää sisällään. Tämän hetken mobiiliratkaisut ovat lähes yksinomaan sovellusratkaisuja, mikä luo rajoitteita alustan sovellusta kehittäessä.

### 7.1 Sivuston kehitys

Sivustolla on potentiaalia kehittyä parempaan suuntaan niin käyttäjän kuin hallinnoinnin nä-kökulmasta. Tästä kyseisestä "alpha"-versiosta voidaan kehittää käyttäjäystävällisempi sekä toimivampi ratkaisu. Tämä toteutettaisiin käyttökokemusten perusteella, mikä loisi pohjan kehitystarpeisiin ja jatkokehitykselle.

Tärkein kehityskohde tulee olemaan tietoturva ja se on yksi keskeisin jatkuvan kehitysproses-sin osa-alue. Tulevaisuuden tarpeisiin pohjautuva kehityssuunnitelma olisi hyvä luoda, mikäli kehitysratkaisuissa päädytään menemään automatisoituun järjestelmään ja ihmisten työmää-rä prosessissa vähenee. Mikäli kehitys menee tähän suuntaan, tietoturva aspektin tärkeys nou-see isompaan rooliin ja tulisi huomioida kehitystyötä tehdessä.

### 7.2 Mobiiliratkaisut

Mobiili-mahdollisuuksia on monia, koska mobiili-käyttöjärjestelmissä on suuria eroja. On ole-massa avoimia sekä suljettuja käyttöjärjestelmä-ympäristöjä. Mobiili-sovelluksen luonti olisi syytä aloittaa ensiksi avointen lähdekoodien omaavista käyttöjärjestelmäpohjista, kuten And-roid, Sailfish OS jne. Avointen järjestelmäpohjien avulla pystytään vaivattomasti testaamaan

sovelluksen toimivuutta mobiilialustalla ja havainnoimaan sovelluksen heikkoudet sekä vahvuudet.

Mobiiliratkaisuissa huomioidaan mahdolliset henkilöt, jotka eivät aina ole koneenäärellä. Myyjät yleisesti liikkuvat paljon luomassa asiakaskontakteja ja näin ollen tarve voi olla suuri liikkuvalle ratkaisulle, johon mobiilipohja on luotu. Tämä mahdollistaisi tilauksen tekemisen "tien päällä" tai asiakkaan luona.

Kyseisessä toimeksiannossa sovelluksen kehittäminen tähän tarkoitukseen olisi turhaa pienen käyttäjämäärän takia, mutta yrityksen kasvaessa sovelluksella voisi olla kysyntää ja mahdollisuuksia isomman mittakaavan tarpeisiin.

## 8 Yhteenveto

Tuotoksen perusteella voidaan päätellä, että työssä esitelty toteutustapa on yksi mahdollisista toteutustavoista, joka vastaa asiakkaan toiveita. Työssä nähtävä toteutus on "Alpha"-versio, jonka pohjalta sovelluksen kehittäminen ja hiominen on helppoa. Toteutus- ja kehittymahdollisuuksia on useita. Työn pohjalta asiakas voi tilatessaan pohtia työssä esitettyjä asioita ja päättää itse, minkälainen ratkaisu vastaa parhaiten yrityksen tarpeita tai mihin suuntaan kehitystä halutaan viedä tulevaisuudessa. Tärkeitä asioita työtä tilatessa ovat, miten asiakas haluaa suojautua mahdollisilta "vikatarjouspyynnöiltä". Vikatarjouspyynnöillä tarkoitetaan internetissä tapahtuvaa häirintää. Eli tässä tapauksessa asiaton henkilö tai henkilöt luovat perusteettomia tarjouspyyntöjä ilman minkäänlaista oikeaa tarkoitusta harrastaa liiketoimintaa.

Mielestämme paras ja helpoin ratkaisu tähän ongelmaan on luoda tietokanta, mihin palvelimen järjestelmänvalvoja luo jokaiselle "tarjouspalvelun" asiakkaalle oman tilin, joka varmistaa asiakkaan tiedot ja oikeanmukaisuuden. Sivuston asiakkaan todentamisen voisi tarkistaa palvelimelta joko käyttäen asiakkaan kirjautumistunnuksia tai VPN-tunnelia hyväksikäyttäen. Työssä esitelty toteutusmenetelmä vastaa mielestämme asiakasyrityksen tarpeisiin hyvin.

Opinnäytetyössä ja siihen kuuluvassa demossa päästiin tavoitteisiin ja tuotos saavutti halutun lopputuloksen huolimatta monista esille nousseista haasteista. Työn edetessä opittiin, että kaikki ei aina mene suunnitelmien mukaan ja ongelmatilanteisiin on suhtauduttava innokkaasti, sekä päättäväisesti. Suurimmaksi ongelmaksi havaittiin resursointi ja suunnittelussa aikataulussa pysyminen työssä olevan demototeutuksen aikana syntyneistä ongelmatilanteista ja niiden ratkaisemiseen tarvitusta lisäajasta johtuen.

Haastavimpana osa-alueena työssä koimme siten demon luonnin vähäisestä ohjelmointikokemuksesta johtuen. Jatkuva kommunikaatio tilaajan kanssa havaittiin myös tärkeäksi. Kommunikaatiossa ollut katkos vaikutti aikataulutukseen sekä työssä tehtävien ratkaisujen tekemiseen ja soveltamiseen.

Demo toteutettiin mahdollisen toteutuksen työkaluksi. Demo mahdollisti määrittelyn havainnollistamisen sivulle sekä tietokannalle. Kerätty tieto auttoi ymmärtämään ratkaisujen toimivuuden ja samalla kehittivät ammatillista käytännön kokemusta.

Jatkokehitystyötä voisi jatkaa sekä tietoja hyödyntää tulevilla hankkeilla että hyödyntää luotua tietopohjaa jatkokehityshankkeissa. Työssä luotu pohja on hyvä ja hyödyllinen apu tarjousjärjestelmän kehityksessä. Luotu tarjousjärjestelmän määrittely ja demo auttaa toteut-

tamaan hyödyllisen ja toimivan version sen kohderyhmälle. Kohderyhmä tutkimusta ei ole toteutettu ja siitä pystyisi toteuttamaan empiiristä tutkimusta.

Työssä opimme, miten tarjousjärjestelmä tulisi toteuttaa. Opimme myös, miten tietokanta tulisi rakentaa niin, että se on yhteensopiva sivuston kanssa. Ammatillinen osaaminen kehittyi työn edetessä. Käytetyt lähteet mahdollistivat oikean kehityssunnan löytämisen ja hyvän lopputuloksen.

## Lähteet

CMS Made Simple 1.6 Beginner's Guide, Hauschildt. (s.8)

SQL in a Nutshell, Kline, Kline, Hunt. (s.1)

Dyer, R. 2008. MySQL in a Nutshell. O'Reilly Media.

Valade, J. 2007. PHP & MySQL For Dummies. Wiley.

Doyle, M. 2009. Beginning PHP 5.3. Wiley.

Hauschildt, S. 2010. CMS Made Simple 1.6 Beginner's Guide. Packt Publishing.

Taylor, A. 2013. SQL For Dummies. Wiley.

Beaulieu, A. 2009. Learning SQL. O'Reilly Media.

What is XAMPP.2011. Knowledge. Viitattu 23.12.2013

<http://www.knowledgement.ie/technical/programming/what-is-xampp>

Php mysql\_connect security. 2010. Rook. Viitattu 3.4.2013

<http://stackoverflow.com/questions/3888065/php-mysql-connect-security>

Mysql tables. 2008. Tizag. Viitattu 18.3.2014

<http://www.tizag.com/mysqlTutorial/mysqltables.php>

Php mysql\_connect security. 2010. Rook. Viitattu 3.12.2013

<http://stackoverflow.com/questions/3888065/php-mysql-connect-security>

What is programming. 2013. David Bolton. Viitattu 11.11.2013

<http://cplus.about.com/od/introductiontoprogramming/p/programmers.htm>

The difference between id and class. 2008. Chris Coyier. Viitattu 18.10.2013

<http://css-tricks.com/the-difference-between-id-and-class/>

## Kuvat

Kuva 1: CMS Toimintakaavio .....	9
Kuva 2: Tietokannan toimintakaavio .....	13
Kuva 3: Sisäänkirjautuminen.....	15
Kuva 4: Tilauslomake.....	16

## Liitteet

Liite 1: Sisäänkirjautuminen .....	24
Liite 2: Tilauslomake .....	25
Liite 3: Katelaskuri .....	26
Liite 4: Tietokanta .....	27

Liite 1: Sisäänkirjautuminen

# SISÄÄNKIRJAUTUMINEN

**Käyttäjätunnus**  **Salasana**

JavaScript-ilmoitus

Yritystiedot:  
Karin Kone  
Mankkaamäki 21 A  
Espoo 02340  
044 0000 000  
kari.pekka@karinkone.fi



Liite 2: Tilauslomake

## Tilauslomake

Katelasikuri

Asiakkaan tiedot:

Karin Kone  
Mankkaamäki 21 A  
Espoo 02340  
044 0000 000  
kari.pekka@karinkone.fi

Tilaus:

Suurkone mallit

Raptor ▼

Suurkone mallit

Cobram AR ▼

Suurkone mallit

Cobram 60 RS ▼

Osat

Osa 4 ▼

Osat

Osa 5 ▼

Osat

Osa 4 ▼

Lisätiedot:

Lähetä

Liite 3: Katelaskuri

---

Hinta  Laske

Hinta	\$1,100.00
15%	\$1,265.00
20%	\$1,320.00
25%	\$1,375.00
30%	\$1,430.00
35%	\$1,485.00
40%	\$1,540.00
45%	\$1,595.00
50%	\$1,650.00
55%	\$1,705.00
60%	\$1,760.00
65%	\$1,815.00
70%	\$1,870.00
75%	\$1,925.00
80%	\$1,980.00
85%	\$2,035.00
90%	\$2,090.00
95%	\$2,145.00

